Remarks

In response to the Office Action dated August 18, 2006 (hereinafter, the "Action"), Applicant respectfully requests reconsideration based on the following remarks.

A review of the claims indicates that:

Claims 1-11, 13-15, and 17-28 were previously pending.

Claims 1-6, 8, 10, 11, 13, 17-20, and 22-28 have been amended.

Claims 7 and 21 have been canceled. Claims 12, 16, and 29-39 were previously canceled.

Claim 40 has been added.

Claims 1-6, 8-11, 13-15, 17-20, 22-28, and 40 are currently pending.

Applicant respectfully submits that the claims as presented are in condition for allowance.

Claim Rejections Under 35 U.S.C. §112

The Office rejects claims 1, 13, and 22 under 35 U.S.C. §112, first paragraph, as failing to comply with the enablement requirement. Specifically, the Office states that "support can not be seen in the specification for updating the association between an executable feature and a graphics element" (Action, pg 3). The Applicant respectfully traverses these rejections.

Without conceding the propriety of the stated rejections, and solely to advance the prosecution of this matter, the Applicant submits the following comments regarding independent claims 1, 13, and 22 to overcome the §112 rejection.

Claim 1

The Applicant submits that "...dynamically updating the information related to the state of operation of the software application and the association in the map data structure upon execution of the executable feature," of amended claim 1, is fully supported under 35 U.S.C. § 112, 1st paragraph, at least by page 5, lines 7-8, page 21, lines 15-24, and page 22 of the Applicant's specification. Applicant submits sufficient specification support exists to overcome the Office's rejection.

Claim 13

The Applicant submits that "...a dynamic updating of the information related to the association and the state of operation of the software application in the map data structure upon the execution of the selected executable feature," in amended claim 13, is fully supported under 35 U.S.C. § 112, 1st paragraph at least by page 5, lines 7-8, page 21 lines 15-24, and page 22 of the Applicant's specification. Applicant submits sufficient specification support exists to overcome the Office's rejection.

Claim 22

The Applicant submits claim 22 does not contain language regarding "...updating the association between a executable feature and a graphics element"

(Action, pg. 3), and therefore specification support for this language is unnecessary.

Accordingly, claims 1, 13, and 22 comply with the 35 U.S.C. §112, first paragraph requirement, and Applicant requests that the §112 rejection be withdrawn.

Claim Rejection Under 35 USC §103(a)

The Office rejects claims 1-6 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,130,911 to Parker et al. (hereinafter "Parker"). The Applicant respectfully traverses this rejection.

Claim 1

Turning to **independent Claim 1**, without conceding the propriety of the stated rejection, and without conceding that Parker provides the teaching for which it was cited in the Action, the Applicant has amended claim 1 as indicated above. For convenience of discussion, the Applicant reproduces here claim 1 as it would stand after entry of the above revisions:

> "A method for testing, the method comprising:
>
> retrieving information descriptive of a state of operation of a software application being tested and at least one graphics element rendered during deterministic execution of the software application being tested, wherein the information identifies an executable feature associated with the at least one graphics element;
>
> storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application in a map data structure containing information related to at least one graphics

> element for testing, the association and information being stored in the map data structure during execution of the software application being tested;
>
> selecting an executable feature from the map data structure based on the association stored in the map data structure, wherein selecting the executable feature proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing;
>
> executing the selected executable feature associated with the graphics element; and
>
> dynamically updating the information related to the state of operation of the software application and the association in the map data structure upon execution of the executable feature."

The Applicant submits that the above revisions to claim 1 are **fully supported under 35 U.S.C. § 112, 1st paragraph**, at least by page 12, line 20 to page 13, line 1; page 16, lines 10-15 and 22-23; page 19, lines 17-23; and page 20, lines 1-4 and lines 10-15, as well as by Figures 4-7 of the Applicant's Specification.

The Parker reference pertains generally to the automated testing of both new and revised computer application programs (Parker, Abstract). More specifically, Parker pertains to graphical user interface testing through the use of a test script written in a high level programming language (Parker, Abstract). The test script is directed towards operation on logical objects, and is not graphical user interface-specific (Parker, Col. 4, lines 10-14).

The Applicant submits that Parker does not teach or suggest every element of Applicant's claim 1. For example, Parker does not disclose "...retrieving information descriptive of a state of operation of a software application being tested and at least one graphics element rendered during deterministic execution of the software application being tested, wherein the information identifies an executable feature associated with the at least one graphics element." Parker also does not disclose "...storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application in a map data structure," the "information being stored... during execution of the software application being tested," as in Applicant's claim 1. The Office cites, inter alia, Col. 4 lines 1-26; Col. 16, line 53 through Col. 17, line 12;Col. 17, lines 2-7; Col. 25, lines 4-8; and Col. 9, lines 50-67 and 59-64. However, Parker does not teach retrieving information descriptive of a state of operation of a software application or a graphics element and associated executable feature of that graphics element, "*rendered during deterministic execution of the software application being tested*," as recited in Applicant's claim 1.

Turning to Parker in more detail, rather than retrieving this information during testing, Parker appears to retrieve testing-related information before testing. More specifically, Parker describes the utilization of test scripts containing high-programming level written commands for the simulation of user events in each testing of the application (Parker, Col. 4, lines 8-10). The test scripts contain "...the control and data structures necessary to validate the GUI's, [as well as] the application program's responses to the input" (Parker, Col. 4, lines 10-12). Because Parker describes a test script containing the graphical user interface (GUI) events to be simulated, the controls necessary to accomplish these simulations, and the GUI response to the simulated events, and because these test script commands are decoded and carried out by Parker's test executive and test driver components (Col. 4, lines 14-44), Parker does not teach "...retrieving information descriptive of a state of operation of a software application being tested and at least one graphics element rendered *during deterministic execution of the software application being tested*, wherein the information identifies an executable feature associated with the at least one graphics element," as in Applicant's claim 1.

Further, Parker does not teach "...storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application in a map data structure," the "information being stored... during execution of the software application being tested," as recited in Applicant's claim 1. The Office cites, inter alia, Parker, Col. 17, lines 2-7 for this feature (Action, page 4). But Parker only describes the "[m]apping between high-level logical object names and actual runtime GUI object names" (Parker, Col. 17, lines 3-4). In this section and the immediately surrounding description, Parker describes a "test tool" having the task of discovering a GUI object's location and state on the specific GUI being tested. (Parker, Col. 16, lines 60 to Col. 17, line 1). The "test tool" in Parker identifies GUI-specific objects from the GUI-generic test script (Parker, Col. 16, lines 53-62). The "mapping" cited by the Office in Col. 17, lines 2-7, refers to the "decoding" of the test script logical language by the "test tool," and the association made between the GUI-specific object and the correlating GUI-generic "logical object name" of the test script. Therefore, this "mapping" in Parker does not teach the "map data structure" for "...storing information related to an association between the executable feature and the at least one graphics element and the state of operation of the software application," the "information being stored... during execution of the software application being tested," as in Applicant's claim 1. Accordingly, claim 1 is allowable over the Parker for at least these reasons, and Applicant respectfully requests that the §103 rejection be withdrawn.

Parker also does not teach "...dynamically updating the information related to the state of operation of the software application and the association in the map data structure upon execution of the executable feature," as in Applicant's claim 1. The Office cites Col. 9, lines 50-67 for this feature (Action, page 4). The Office also states "Parker doesn't explicitly state the updating of an association between an (LSE) and a (PSE), but he does teach allowing 'the LSEM to create, modify and destroy specific LSEs,'" and that "(o)ne would have been motivated to make such a combination because this allows the same script to be used to test several similar screen elements" (Action, page 4). Applicant respectfully disagrees.

As disclosed in claim 1, the "deterministic mode" utilized in a specific testing process run is "for a systematic order of software application execution during testing." Thus, the selection and execution of each executable feature associated with a graphics element proceeds according to the deterministic mode being used, *not* through the use of a script. Therefore, the motivation stated in the Action to use "the same script to be used to test several similar screen elements," is not demonstrated, as the method of claim 1 utilizes a comprehensive deterministic mode, not a test script, for the testing of a software application. Accordingly, claim 1 is allowable over Parker for at least these reasons, and Applicant respectfully requests that the §103 rejection be withdrawn.

The Applicant submits that Parker also does not teach or suggest "...selecting an executable feature from the map data structure based on the association stored in the map data structure, wherein selecting the executable feature proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing," as in Applicant's amended claim 1. Instead, Parker teaches the decoding and execution of pre-written test script commands, where the test script commands recite the GUI events to be simulated, the controls necessary to accomplish the simulations, and the GUI response to the recreations (Col. 4, lines 14-44). The test scripts contain a set of commands to be simulated on a GUI, but these tests do not represent a comprehensive testing of the entire software application; rather the test scripts test a pre-determined set of GUI actions to assess performance. As such, Parker does not teach the selection of an executable feature, during the testing execution of a software application, according to a sequence determined by a utilized deterministic code.

The Parker reference does not teach or suggest each element of Applicant's claim 1. Therefore, Applicant respectfully submits that Parker does not support a §103 rejection of claim 1. The Applicant thus requests reconsideration and withdrawal of the § 103 rejection of claim 1.

Claims 2-6

Claims 2-6 depend from independent claim 1. Therefore, the comments directed above to claim 1 apply equally to claims 2-6 as well as for the additional features recited in these dependent claims.

For at least these reasons set forth with respect to amended claim 1, Applicant submits that dependent claims 2-6 are also not obvious over Parker.

Claims 2-6 depend from claim 1, and therefore, these claims should be allowable for the reasons stated above with respect to claim 1.

Claim Rejection Under 35 USC §103(a)

Claims 7-11, 13-15, and 17-28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Parker in view of U.S. Patent No. 6,415,396 to Singh et al. (hereinafter "Singh"). Applicant respectfully traverses the rejection.

Singh generally pertains to generating and maintaining a regression test case of a system set directly and automatically from requirement models (Singh, Abstract). As features of a system are modified over time, a regression test case is automatically regenerated to test the system with the modifications (Singh, Summary).

Claims 7-11

Claim 7 has been canceled, therefore the Office's rejection of this claim is now moot.

Claims 8-11 are allowable by virtue of their dependency upon independent claim 1 which is allowable over Parker for at least the reasons described above in response to the §103 rejection of claim 1. Claims 8-11 are also allowable over the Parker-Singh combination because Singh does not address the deficiencies of Parker, specifically, the Singh reference does not teach "...selecting an executable feature from the map data structure based on the association stored in the map data structure, wherein selecting the executable feature proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing," as recited in Applicant's claim 1.

Singh also does not teach "...selecting executable features in a depth-first mode of operation," as recited in Applicant's claim 10 and "...selecting executable features in a breadth-first mode of operation," as recited in Applicant's claim 11. The Office cites Col. 3, lines 47-51 and Col. 12, lines 50-63, and Col. 13, lines 50-63 of the Singh reference for this feature (Action, page 7). However, Singh does not teach or suggest a comprehensive process for testing that includes a sequence for the selection of executable features that is determined by a deterministic mode, as recited in claims 1, 10, and 11.

Singh only describes the creation of regression test cases where the selection techniques used to produce the set of test cases are applied to a graph that represents a model of the system being tested (Singh, Col. 3, lines 47-50). Singh discloses that performances of regression tests are limited by schedule constraints and access issues, and that the "optimal set of test cases is that set which provides coverage of the graph and is minimal with respect to the number of tests and test length" (Singh, Col. 3, lines 51-56). The creation of an "optimal" set of test cases constructed to represent test coverage of an entire system does not teach a comprehensive process for testing that includes a sequence for the selection of executable features determined by a deterministic mode, as recited in claims 1, 10, and 11.

Additionally, Singh does not teach the use of deterministic modes for comprehensive testing of a software application, as described in Applicant's claims 1, 10, and 11. Instead, Singh describes "depth-first" and "breadth-first" searches for identifying the nodes of the model of the system (Singh, Col. 13, lines

50-64). The described searches in Singh are performed to suggest potential repair paths between nodes of a system model, and are performed "...until enough potential repair paths are reported" (Singh, Col. 13, line 50 through Col. 14, line 3). This description in Singh does not teach "...selecting the executable feature...according to a sequence determined by one of a plurality of deterministic modes that ensures a systematic order to software application execution during testing," or "...selecting executable features in a depth-first mode [or breadth-first mode] of operation," as recited in Applicant's claims 1, 10, and 11.

Because the Parker-Singh combination does not teach or suggest each element of claims 8-11, Applicant respectfully submits that the Parker and Singh references do not support a §103 rejection of claims 8-11. The Applicant thus requests reconsideration and withdrawal of the § 103 rejection of claims 8-11.

Claim 13

Turning next to **independent Claim 13**, without conceding the propriety of the stated rejection, and without conceding that Parker and Singh provide the teaching for which they were cited in the Action, the Applicant has amended claim 13 as indicated above. For convenience of discussion, the Applicant reproduces here claim 13 as it would stand after entry of the above revisions:

> "A system for generating a map, comprising:
>
> a capture agent for retrieving information descriptive of a state of operation of a software application being tested and a plurality of graphics elements rendered during deterministic execution of the software application, the information including an executable feature associated with each graphics element;
>
> an application driver for storing information in a map data structure related to an association between each executable feature and corresponding graphics element and a state of operation of the software

application during execution of the software application being tested, wherein the map data structure contains information related to at least one graphics element for testing;

an application driver for deterministically selecting one of the executable features stored in the map data structure based on the information stored in the map data structure, wherein deterministically selecting proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing;

a command agent for executing the selected executable feature; and

an indicator for tracking a dynamic updating of the information related to the association and the state of operation of the software application in the map data structure upon the execution of the selected executable feature."

The Applicant submits that the above revisions to claim 13 are fully supported under 35 U.S.C. § 112, 1st paragraph, at least by page 5, lines 7-8; page 12, line 20 to page 13, line 1; page 16, lines 10-15 and 22-23; page 19, lines 17-23; page 20, lines 1-4 and lines 10-15, and page 21 lines 15-19, as well as by Figures 4-7 of the Applicant's Specification.

Claim 13 is rejected under the same rationale, and with the same citations of the Parker and Singh references, as that in the rejection of claims 1 and 8-11. (Action, p. 8). As described above in the response to the rejection of claim 1, Parker does not teach or suggest many of the features recited in claim 13. Further, the Singh reference does not address the deficiencies of Parker, as described above in the response to the rejection of claims 8-11. Specifically, the Parker and Singh references do not:

- teach or suggest retrieving information descriptive of a state of operation of a software application being tested and a plurality of graphics elements rendered during deterministic execution of the software application;

- teach or suggest storing an association between an executable feature and corresponding graphics element, and a state of operation of the software application, during execution of the software application being tested; or
- teach or suggest deterministically selecting an executable feature according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing.

The Parker-Singh combination does not teach or suggest the features recited in claim 13. Accordingly, claim 13 is allowable over Parker-Singh for at least these reasons, and Applicant respectfully requests that the §103 rejection be withdrawn.

Claims 14, 15, 17-21

Claim 21 has been canceled, and therefore the Office's rejection of this claim is now moot.

Claims 14, 15, 17-20 depend from independent claim 13. Therefore, the comments directed above to claim 13 apply equally to claims 14, 15, 17-20 as well as for the additional features recited in these dependent claims.

For at least these reasons set forth with respect to amended claim 13, Applicant submits that dependent claims 14, 15, 17-20 are also not obvious over Parker in view of Singh. Claims 14, 15, 17-20 depend from claim 13, and therefore, these claims should be allowable for the reasons stated above with respect to claim 13.

Claim 22

Turning next to **independent Claim 22**, without conceding the propriety of the stated rejection, and without conceding that Parker and Singh provide the teaching for which they were cited in the Action, the Applicant has amended claim 22 as indicated above. For convenience of discussion, the Applicant reproduces here claim 22 as it would stand after entry of the above revisions:

> "A method for systematically invoking an executable feature of a software application having a graphical user interface, the method comprising:
>
> retrieving information descriptive of a state of operation of a software application being tested and at least one graphics element rendered during deterministic execution of the software application, the information including an executable feature associated with the at least one graphics element;
>
> storing information related to an association between the executable feature and corresponding graphics element and the state of operation of the software application in a map data structure to contain information related to at least one graphics element for testing, the association and information being stored in the map data structure during execution of the software application;
>
> selecting from the map data structure at least one executable feature associated with a graphics element that has not been previously executed, wherein selecting the at least one executable feature proceeds according to a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing; and
>
> executing the selected at least one executable feature."

The Applicant submits that the above revisions to claim 22 are fully supported under 35 U.S.C. § 112, 1st paragraph, at least by page 5, lines 7-8; page 12, line 20 to page 13, line 1; page 16, lines 10-15 and 22-23; page 19, lines 17-

23; page 20, lines 1-4 and lines 10-15, and page 21 lines 15-19, as well as by Figures 4-7 of the Applicant's Specification.

Claim 22 is rejected under a similar rationale and support as described in the rejections of claim 1 and claim 13 (Action, pgs. 10-12). As described above in the response to the rejection of claim 1, Parker does not teach or suggest many of the features recited in claim 22. Further, the Singh reference does not address the deficiencies of Parker, as described above in the response to the rejection of claims 8-11. Specifically, the Parker and Singh references do <u>not</u>:

- teach or suggest retrieving information descriptive of a state of operation of a software application being tested and a plurality of graphics elements rendered during deterministic execution of the software application;
- teach or suggest storing an association between an executable feature and corresponding graphics element, and a state of operation of the software application, during execution of the software application being tested;
- teach or suggest selecting from the map data structure an executable feature associated with a graphics element that has not been previously executed; or
- teach or suggest selecting an executable feature in a sequence determined by one of a plurality of deterministic modes for a systematic order of software application execution during testing.

The Parker-Singh combination does not teach or suggest the features recited in claim 22. Accordingly, claim 22 is allowable over Parker-Singh for at least these reasons, and Applicant respectfully requests that the §103 rejection be withdrawn.

Claims 23-28

Claims 23-28 depend from independent claim 22. Therefore, the comments directed above to claim 22 apply equally to claims 23-28 as well as for the additional features recited in these dependent claims.

For at least these reasons set forth with respect to amended claim 22, Applicant submits that dependent claims 23-28 are also not obvious over Parker in view of Singh. Claims 23-28 depend from claim 22, and therefore, these claims are allowable for the reasons stated above with respect to claim 22.

Conclusion

The Applicant submits that the claims in this application are now in condition for allowance. Applicant respectfully requests that an early Notice of Allowability be issued. If there are any outstanding issues that would prevent favorable action on this application, Applicant respectfully requests that the undersigned attorney be contacted for the purpose of scheduling an interview.

Respectfully Submitted,

Dated: 11/18/06 By: ______________________

Emily H. Ling
Lee & Hayes, pllc
Reg. No. 58,619
(509) 324-9256 ext. 222